Title

varwle — Obtain Wald lag-exclusion statistics after var or svar

Syntax
Menu
Description
Options
Remarks and examples
Stored results
Methods and formulas
References
Also see

Syntax

```
varwle [, estimates(estname) separator(#)]
```

varwle can be used only after var or svar; see [TS] **var** and [TS] **var svar**.

Menu

Statistics > Multivariate time series > VAR diagnostics and tests > Wald lag-exclusion statistics

Description

varwle reports Wald tests the hypothesis that the endogenous variables at a given lag are jointly zero for each equation and for all equations jointly.

Options

estimates(*estname*) requests that varwle use the previously obtained set of var or svar estimates stored as *estname*. By default, varwle uses the active estimation results. See [R] **estimates** for information on manipulating estimation results.

separator(*#*) specifies how often separator lines should be drawn between rows. By default, separator lines do not appear. For example, separator(1) would draw a line between each row, separator(2) between every other row, and so on.

Remarks and examples

After fitting a VAR, one hypothesis of interest is that all the endogenous variables at a given lag are jointly zero. varwle reports Wald tests of this hypothesis for each equation and for all equations jointly. varwle uses the estimation results from a previously fitted var or svar. By default, varwle uses the active estimation results, but you may also use a stored set of estimates by specifying the estimates() option.

If the VAR was fit with the small option, varwle also presents small-sample F statistics; otherwise, varwle presents large-sample chi-squared statistics.

▷ Example 1: After var

We analyze the model with the German data described in [TS] **var** using `varwle`.

```
. use http://www.stata-press.com/data/r13/lutkepohl2
(Quarterly SA West German macro data, Bil DM, from Lutkepohl 1993 Table E.1)
. var dln_inv dln_inc dln_consump if qtr<=tq(1978q4), dfk small
  (output omitted)
. varwle
  Equation: dln_inv
```

lag	F	df	df_r	Prob > F
1	2.64902	3	66	0.0560
2	1.25799	3	66	0.2960

```
  Equation: dln_inc
```

lag	F	df	df_r	Prob > F
1	2.19276	3	66	0.0971
2	.907499	3	66	0.4423

```
  Equation: dln_consump
```

lag	F	df	df_r	Prob > F
1	1.80804	3	66	0.1543
2	5.57645	3	66	0.0018

```
  Equation: All
```

lag	F	df	df_r	Prob > F
1	3.78884	9	66	0.0007
2	2.96811	9	66	0.0050

Because the VAR was fit with the `dfk` and `small` options, `varwle` used the small-sample estimator of $\widehat{\Sigma}$ in constructing the VCE, producing an F statistic. The first two equations appear to have a different lag structure from that of the third. In the first two equations, we cannot reject the null hypothesis that all three endogenous variables have zero coefficients at the second lag. The hypothesis that all three endogenous variables have zero coefficients at the first lag can be rejected at the 10% level for both of the first two equations. In contrast, in the third equation, the coefficients on the second lag of the endogenous variables are jointly significant, but not those on the first lag. However, we strongly reject the hypothesis that the coefficients on the first lag of the endogenous variables are zero in all three equations jointly. Similarly, we can also strongly reject the hypothesis that the coefficients on the second lag of the endogenous variables are zero in all three equations jointly.

If we believe these results strongly enough, we might want to refit the original VAR, placing some constraints on the coefficients. See [TS] **var** for details on how to fit VAR models with constraints.

◁

▷ Example 2: After svar

Here we fit a simple SVAR and then run `varwle`:

```
. matrix a = (.,0\.,.)

. matrix b = I(2)

. svar dln_inc dln_consump, aeq(a) beq(b)
Estimating short-run parameters

Iteration 0:   log likelihood = -159.21683
Iteration 1:   log likelihood =  490.92264
Iteration 2:   log likelihood =  528.66126
Iteration 3:   log likelihood =  573.96363
Iteration 4:   log likelihood =  578.05136
Iteration 5:   log likelihood =  578.27633
Iteration 6:   log likelihood =  578.27699
Iteration 7:   log likelihood =  578.27699

Structural vector autoregression

 ( 1)  [a_1_2]_cons = 0
 ( 2)  [b_1_1]_cons = 1
 ( 3)  [b_1_2]_cons = 0
 ( 4)  [b_2_1]_cons = 0
 ( 5)  [b_2_2]_cons = 1

Sample:  1960q4 - 1982q4                           No. of obs      =        89
Exactly identified model                           Log likelihood  =   578.277
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
/a_1_1	89.72411	6.725107	13.34	0.000	76.54315	102.9051
/a_2_1	-64.73622	10.67698	-6.06	0.000	-85.66271	-43.80973
/a_1_2	0	(constrained)				
/a_2_2	126.2964	9.466318	13.34	0.000	107.7428	144.8501
/b_1_1	1	(constrained)				
/b_2_1	0	(constrained)				
/b_1_2	0	(constrained)				
/b_2_2	1	(constrained)				

The output table from `var svar` gives information about the estimates of the parameters in the **A** and **B** matrices in the structural VAR. But, as discussed in [TS] **var svar**, an SVAR model builds on an underlying VAR. When `varwle` uses the estimation results produced by `svar`, it performs Wald lag-exclusion tests on the underlying VAR model. Next we run `varwle` on these `svar` results.

```
. varwle
```

Equation: dln_inc

lag	chi2	df	Prob > chi2
1	6.88775	2	0.032
2	1.873546	2	0.392

Equation: dln_consump

lag	chi2	df	Prob > chi2
1	9.938547	2	0.007
2	13.89996	2	0.001

Equation: All

lag	chi2	df	Prob > chi2
1	34.54276	4	0.000
2	19.44093	4	0.001

Now we fit the underlying VAR with two lags and apply `varwle` to these results.

```
. var dln_inc dln_consump
```
(*output omitted*)

```
. varwle
```

Equation: dln_inc

lag	chi2	df	Prob > chi2
1	6.88775	2	0.032
2	1.873546	2	0.392

Equation: dln_consump

lag	chi2	df	Prob > chi2
1	9.938547	2	0.007
2	13.89996	2	0.001

Equation: All

lag	chi2	df	Prob > chi2
1	34.54276	4	0.000
2	19.44093	4	0.001

Because `varwle` produces the same results in these two cases, we can conclude that when `varwle` is applied to `svar` results, it performs Wald lag-exclusion tests on the underlying VAR.

◁

Stored results

`varwle` stores the following in `r()`:

Matrices

if `e(small)==""`

`r(chi2)`	χ^2 test statistics
`r(df)`	degrees of freedom
`r(p)`	p-values

if `e(small)!=""`

`r(F)`	F test statistics
`r(df_r)`	numerator degrees of freedom
`r(df)`	denominator degree of freedom
`r(p)`	p-values

Methods and formulas

`varwle` uses `test` to obtain Wald statistics of the hypotheses that all the endogenous variables at a given lag are jointly zero for each equation and for all equations jointly. Like the `test` command, `varwle` uses estimation results stored by `var` or `var svar` to determine whether to calculate and report small-sample F statistics or large-sample chi-squared statistics.

Abraham Wald (1902–1950) was born in Cluj, in what is now Romania. He studied mathematics at the University of Vienna, publishing at first on geometry, but then became interested in economics and econometrics. He moved to the United States in 1938 and later joined the faculty at Columbia. His major contributions to statistics include work in decision theory, optimal sequential sampling, large-sample distributions of likelihood-ratio tests, and nonparametric inference. Wald died in a plane crash in India.

References

Amisano, G., and C. Giannini. 1997. *Topics in Structural VAR Econometrics*. 2nd ed. Heidelberg: Springer.

Hamilton, J. D. 1994. *Time Series Analysis*. Princeton: Princeton University Press.

Lütkepohl, H. 1993. *Introduction to Multiple Time Series Analysis*. 2nd ed. New York: Springer.

Mangel, M., and F. J. Samaniego. 1984. Abraham Wald's work on aircraft survivability. *Journal of the American Statistical Association* 79: 259–267.

Wolfowitz, J. 1952. Abraham Wald, 1902–1950. *Annals of Mathematical Statistics* 23: 1–13 (and other reports in same issue).

Also see

[TS] **var** — Vector autoregressive models

[TS] **var svar** — Structural vector autoregressive models

[TS] **varbasic** — Fit a simple VAR and graph IRFs or FEVDs

[TS] **var intro** — Introduction to vector autoregressive models